# Robust Grid Computing using Peer-to-Peer Services

A. Sussman, P. Keleher, B. Bhattacharjee, D. Richardson, D. Wellnitz

## Personnel

- Alan Sussman - PI - Grid computing
- Pete Keleher - co-I - P2P algorithms
- Bobby Bhattacharjee - co-I - P2P algorithms
- Derek Richardson - co-I - astronomy applications
- Dennis Wellnitz - co-I - Deep Impact applications
- Michael Marsh - research scientist - implementation and supervising student implementation work
- Jik-Soo Kim - graduate student - matchmaking algorithms
- Jaehwan Lee - graduate student - implementation, matchmaking algorithms for multi-processor/multi-core machines
- Sukhyun Song - graduate student - implementation, peer security issues
- Beomseok Nam - graduate student/postdoc - basic P2P algorithms and implementation
- San Ratanasanya - visiting graduate student (from Thailand) - client implementation

## Activities and Findings

Work in the second year of the project has concentrated on algorithms and simulations to create a fully functional peer that both performs matchmaking well, and scales to large numbers of machines with no serious load imbalances. Such load imbalances can arise from both problems in the initial matchmaking process that assigns jobs to machines, and from maintaining the structured peer-to-peer (P2P) network. In work that was published in the 2007 High Performance Distributed Computing (HPDC) conference proceedings, we described and evaluated algorithms for significantly improving load balance in the matchmaking process, compared to our earlier algorithms. The improved algorithms still guarantees that a job will be assigned to a machine that meets the job's minimum resource requirements, but also employs distributed load balancing techniques that piggyback load information onto the messages used to maintain the P2P network, and change the algorithm for then assigning a job to a node to be run. This is all still within the context of a completely decentralized system, with no global information available at any single point in the system.

Additional algorithm and simulation work has focused on providing support for *categorical* resource types, meaning ones that require either an exact match between job and compute node (e.g., system architecture, such as IA-32 or PowerPC) or ones that allow a range of discrete values for the match (e.g., a relatively new version of an operating system, such as Linux kernel version 2.4 or higher). In a paper submitted to the 2008 Grid Computing conference, we describe techniques based on *virtual peers* and a transformation technique based on space-filling curves to limit the number of categorical dimensions in the multi-dimensional P2P network (based on a structured P2P network know as a Content Addressable Network, or CAN, as described in last year's report). Optimizations that limit both the size and number of messages then required to maintain the resulting CAN are also needed, and described in that paper.

Without those optimizations, our simulations showed that some nodes in the P2P network could experience very high loads in maintaining the network and in performing matchmaking, but with the optimizations all of those concerns are addressed. Overall, our findings from this work are that, with all the mechanisms in place, the P2P grid system can effectively match jobs to machines with good load balance, and with low overall system overhead.

## Implementation status

Much effort this year has also gone into the implementation of a usable software system. Based on the results of simulations of our initial algorithms from year 1, and the new ones from this year, we have built our first peer implementation. Initial versions of the peer software have been thoroughly tested, and a large scale evaluation with the astronomy collaborators on the project is under way. The evaluation involves running the peer software on multiple clusters and desktop machines in both the Computer Science and Astronomy departments at Maryland, on a total of well over 100 machines, and running thousands of astronomy simulations (still being determined by the astronomers) through the complete system. We intend to analyze many aspects of the system behavior through extensive logging of the peer behavior during the experiment, including overall scalability and how well our system simulations match the behavior of the real system. We hope to report on our analysis of the results of the evaluation at the AISR PI meeting in May.

We have also implemented client software to enable scientists to submit jobs into a grid, bootstrapping software to ease setting up a grid and simplify adding machines into a grid (including scripts for adding a large set of machines in a cluster at one time). We also have created software with a simple GUI to make it easier for users to write job specifications for submission with the client software. Finally, authentication mechanisms have been incorporated into the peer and client software, currently based on SSH keys, to validate that a user is authorized to either add a machine as a peer to a grid, or to submit a job into a grid. Several security mechanism have also been incorporated into the peer software, to protect a host machine from malicious behavior by a job running on the machine (either intentional or accidental). Such mechanisms can be optionally enabled at peer startup time on a host computer. A short paper on the overall status of the algorithms and implementation will appear in the proceedings of the 2008 NSF Next Generation Systems Software workshop, to be held in April at the International Parallel and Distributed Processing Symposium.

## Training and Development

The project has provided training for the graduate students on the project, getting them working with desktop grid and peer-to-peer technology. One student will finish his PhD dissertation this summer or fall on work done for the project, and two other PhD students are currently working on the project, both on the peer implementation and on research topics related to the peer algorithms. The graduating student will be presenting his dissertation work as a poster in the PhD symposium at the International Parallel and Distributed Processing Conference in April 2008. The project has also brought together faculty in Computer Science and Astronomy with different skills and expertise. The computer scientists are learning about the astronomy applications and their requirements, and are now starting to mine the information collected from using the peers in a grid to run the astronomy applications. This information will be used to further tune and optimize the peer algorithms and implementation. The astronomers are also learning what the computer science tools can do to help them share resources with their colleagues within Maryland and at other institutions. The collaboration among the computer science faculty has been very fruitful, sharing expertise about peer-to-peer techniques and Grid computing that no one person has.

The project inspired the PI to offer a graduate class on the intersection of Grid and Peer-to-Peer computing in Spring 2007, which trained both computer science graduate students and computational science graduate students in the Maryland Applied Math program in the technologies and potential applications. The cross fertilization between computer science and computational science students proved very fruitful for both sides, especially since they did significant programming projects together.

# Publications

## Journal

1. J.-S. Kim, B. Nam, P. Keleher, M. Marsh, B. Bhattacharjee and A. Sussman. "Trade-offs in Matching Jobs and Balancing Load for Distributed Desktop Grids", to appear in *Future Generation Computer Systems – International Journal of Grid Computing: Theory, Methods & Applications*, Vol. 24, No. 5, 2008.

## Conference

1. J.-S. Kim, B. Nam, M. Marsh, P. Keleher, B. Bhattacharjee and A. Sussman. "Integrating Categorical Resource Types into a P2P Desktop Grid System", submitted to *International Conference on Grid Computing (Grid 2008)* , April 2008.

2. M. Marsh, J.-S. Kim, B. Nam, J. Lee, S. Ratanasanya, B. Bhattacharjee, P. Keleher, D. Richardson, D. Wellnitz and Alan Sussman. "Matchmaking and Implementation Issues for a P2P Desktop Grid", to appear in *Proceedings of the 2008 NSF Next Generation Software Workshop*, April 2008.

3. J.-S. Kim, P. Keleher, M. Marsh, B. Bhattacharjee and A. Sussman. "Using Content-Addressable Networks for Load Balancing in Desktop Grids", *Proceedings of the 16th IEEE International Symposium on High Performance Distributed Computing (HPDC-16)*, June 2007

4. J.-S. Kim, B. Nam, M. Marsh, P. Keleher, B. Bhattacharjee, D.C. Richardson, D. Wellnitz and A. Sussman. "Creating a Robust Desktop Grid using Peer-to-Peer Services", *Proceedings of the 2007 NSF Next Generation Software Workshop*, March 2007.

5. J.-S Kim, B. Nam, P. Keleher, M. Marsh, B. Bhattacharjee and A. Sussman. "Resource Discovery Techniques in Distributed Desktop Grid Environments", *Proceedings of the 7th IEEE/ACM International Conference on Grid Computing - GRID 2006*, September 2006. Best paper award.

# Future plans

We are continuing to develop the algorithms for matching resource requests to available resources, in particular concentrating on multi-core and multi-processor grid nodes. Effectively utilizing such resources, which are becoming more widespread and important, is an open question that has not been addressed at all in the desktop grid community to date. Another area of ongoing research is in techniques for dynamic load balancing. Our current algorithms and implementation perform matchmaking once, when a job is submitted into a grid. That matchmaking is done using the current (approximate) state of the overall set of peers that exists at that time. We will investigate methods for determining when to initiate dynamic load balancing algorithms, to adapt to the ever changing grid environment. We will also investigate the behavior of both push and pull algorithms for dynamic load balancing, with pull algorithms designed to pull jobs to underloaded peers, and push algorithms designed to push jobs away from overloaded peers.

The algorithms for dealing with categorical resource types are being implemented in the peer software, and will be thoroughly tested and evaluated in the upcoming year. We are in the process of fully testing and deploying the software within the project, both in computer science and to the astronomy collaborators. Large scale testing of the peer in a distributed grid system is under way, and will consume the majority of the project resources in the coming, final year of the project. We will evaluate the reliability and scalability of the algorithms and implementation under real workloads, first from within the project with astronomy applications run by the astronomy co-Is on the project, and then in a wider deployment within the wider computational science community at Maryland (through the Institute for Advanced Computer Studies - UMIACS), and then to collaborators outside Maryland (especially PI Sussman's space science collaborators from the space weather modeling community).

We will also continue to work on characterizing the different types of workloads that the system is exposed to, to be able to simulate the behavior of the system under such workloads, for much larger resource configurations than are likely to become available to the project. We have done some mining of Condor logs both from within clusters currently running in the Maryland Institute for Advanced Computer Studies (UMIACS) that serve a diverse community of computational scientists, and on a cluster run by Astronomy co-I Richardson that is used for many types of astronomy simulations. We have also obtained logs from large Condor clusters at the University of Notre Dame and Purdue University, through a computer science colleague at Notre Dame, and will analyze those too. Finally, we will perform deep analysis of the logs obtained from the real peer used to run real astronomy and other applications, to both validate the results of our system simulations to date, and to give us confidence that our simulations of the system behavior on very large system configurations (with many more machines than we can possibly obtain for experiments) are accurate and useful in predicting large scale system behavior.